SISTEMA DE RECONHECIMENTO BASEADO EM RANDOM FOREST PARA CARACTERES DE CAPTCHAS

Ademir Rafael Marques Guedes, Victor Luiz Guimarães

Universidade Federal de Ouro Preto(UFOP)
Departamento de Computação

ABSTRACT

O reconhecimento de caracteres em CAPTCHAS é um problema muito estudado na área de Visão Computacional e está diretamente ligado às bases de dados de onde foram extraídos as imagens e às características dessa, sendo que busca-se sempre um sistema que consiga executar esse reconhecimento com grande acurácia. Nesse trabalho, propõe-se um sistema de reconhecimento dos caracteres usando como classificador o método Random Forest e como extratores de características o método de histogramas orientados e características estruturais dos caracteres. O sistema foi testado com 11908 imagens extraídas do sistema de CAPTCHAS utilizado pela Receita Federal.

***Index Terms*—** Reconhecimento de dígitos, Random Forest, HoG, Structural Caracteristics

1. INTRODUÇÃO

O problema de reconhecimento automatizado de caracteres, sejam eles manuscritos ou digitados, através de imagens é um dos problemas mais clássicos de Visão Computacional, sendo de importante valia em diversas áreas, como por exemplo na conversão de um texto impresso para texto digitalizado ou no reconhecimento dos números que identificam as residências de uma rua pelo sistema do Google Street View. Também na classe dos problemas de reconhecimento de caracteres pode-se encontrar especificamente o problema de reconhecimento de CAPTCHAS (Completely Automated Public Turing test to tell Computers and Humans Apart), os famosos códigos que devem ser digitados por um usuário a fim de que ele comprove que é um ser humano e assim obtenha acesso a determinados serviços de um determinado sistema.

O reconhecimento de CAPTCHAS é geralmente um problema complexo, visto que as bases de imagens são geradas utilizando-se diversos artifícios de manipulação das imagens e dos caracteres nelas presentes, com intuito de impedir o reconhecimento automatizado dos mesmos.

Neste trabalho pretende-se apresentar uma análise de dois métodos baseados na extração de características estruturais dos caracteres e de histogramas de gradientes orientados (HoG) e na classificação através do método de Random Forest. Ambos os métodos foram desenvolvidos para o reconhecimento de caracteres previamente segmentados presentes nos CAPTCHAS extraídos da base utilizada nos serviços da Receita Federal do Brasil.

2. PRÉ-PROCESSAMENTO

Nesse trabalho, foram utilizadas imagens contendo, cada uma, um único caracter alfabético ou numérico, segmentado a partir de um conjunto de seis outros que formavam previamente um CAPTCHA da base de imagens dos serviços da Receita Federal.

Como o objetivo dos CAPTCHAS é evitar o reconhecimento dos caracteres presentes nas imagens de forma automatizada, a maior parte das imagens apresenta grande quantidade de ruídos como traços, borrões e distorções que dificultam o processo de classificação. Para uma melhor performance na detecção e classificação dos caracteres presentes em cada imagem, é ideal que se remova a maior parte dos ruídos possível, de forma a obter uma imagem cada vez mais limpa que contenha, preferencialmente somente o caracter a ser reconhecido.

Para a remoção dos ruídos da imagem foram utilizadas técnicas de Processamento de Imagens baseadas nas técnicas de pré-processamento presentes em [1]. Primeiramente, utilizou-se um método de binarização da imagem baseado em limiar, para deixar as imagens com o fundo branco e com os caracteres (e ruídos) em tons pretos. A binarização, além de tornar o trabalho de remoção de ruídos e posterior reconhecimento do caracter mais simples, também realiza por sua vez remoção de ruídos leves como borrões leves e ruídos fracos que se encontram em tons muito próximos do branco ou do preto, pois esses são convertidos para preto ou branco.

Em seguida, percorre-se toda a extensão da imagem à procura de grupos de pixels pretos em sequência de largura ou altura que não ultrapasse três unidades, ou seja, quaisquer riscos compostos por até três pixels pretos. Tais riscos são então removidos da imagem, sendo que essa remoção baseia-se na observação de que os traços que compõe os caracteres variam de 4 a 7 pixels de largura ou altura, ou seja, remove-se boa

parte dos ruídos da imagem e evita-se a remoção de parte útil dos caracteres nesse processo. Um exemplo do método de pré-processamento aplicado às imagens pode ser visto na figura 1.

(a) Imagem original

(b) Imagem binarizada

(c) Imagem pré-processada

Fig. 1. Exemplo das etapas de pré-processamento

3. EXTRAÇÃO DE CARACTERÍSTICAS

Extrair bem as características das imagens de forma que seja possível distingui-las umas das outras de forma fácil e clara é muito importante para que seja possível apresentar um bom resultado na classificação, dado que essa fase depende intrinsecamente desta. Para isso, após um pesquisa na literatura, optou-se por utilizar dois métodos, a saber o HoG e o Strutural Caracteristics.

3.1. Structural Caracteristics

Proposto em [2], o Structural Caracteristics, como o próprio nome diz, tenta extrair características estruturais da imagem. Isso é feito utilizando-se de histogramas e criando-se perfis da mesma.

O método espera uma matriz binária de 32x32. Tendo a posse desta, o algoritmo calcula os histogramas verticais e horizontais. Logo em seguida é calculado o histograma radial da imagem. Nesse trabalho foram utilizados 72 vetores cada um com um deslocamento de 5 quando comparado com o anterior. Além dos histogramas, foi propostos pelos autores a utilização de dois perfis, um que marca a posição dos pixels pretos de dentro para fora partindo do centro da imagem, e outra que marca a posição dos mesmos de fora para dentro, partindo de uma extremidade e rumando para o centro. No total o vetor de características proposto possui 280 posições.

Assim, sendo $F(m, n)$ os valores da intensidade dos pixels da imagem a ser processada define-se o cálculo das características estruturais das imagens através das fórmulas a seguir:

$$H_v(n) = \sum_m F(m, n) \tag{1}$$

$$H_h(m) = \sum_n F(m, n) \tag{2}$$

$$H_r(\theta) = \sum_{i=1}^{16} F(||16 - i\sin\theta||, ||16 + i\cos\theta||) \tag{3}$$
$$\theta = 5 * k, k \in [0, 72]$$

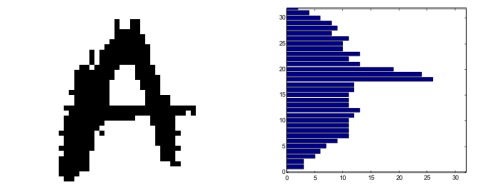
$$P_{oi}(\theta) = I : \sum_{i=16}^{I-1} F(||16 - i\sin\theta||, ||16 + i\cos\theta||) \equiv 0 \tag{4}$$
$$\&F(||16 - I\sin\theta||, ||16 + I\cos\theta||) \equiv 1$$
$$\theta = 5 * k, k \in [0, 72]$$

$$P_{io}(\theta) = J : \sum_{i=0}^{J-1} F(||16 - i\sin\theta||, ||16 + i\cos\theta||) \equiv 0 \tag{5}$$
$$\&F(||16 - J\sin\theta||, ||16 + J\cos\theta||) \equiv 1$$
$$\theta = 5 * k, k \in [0, 72]$$

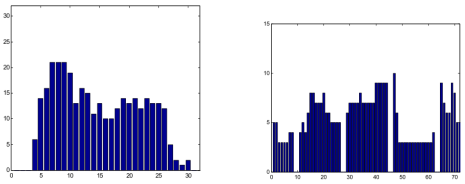
Em 1, temos a definição do histograma vertical, ou seja, a contagem do número de pixels pretos em cada coluna da imagem. Similarmente, em 2, temos a definição do histograma horizontal, em que realiza-se a contagem do número de pixels pretos em cada linha da imagem. Em 3, define-se o histograma radial, em que realiza-se a contagem de pixels pretos em um rad que parte do centro da imagem e termina na borda da imagem, formando um ângulo θ com o eixo horizontal. O histograma radial nesse trabalho foi calculado a cada 5 graus, gerando assim 72 características.

Já em 4, faz-se a definição do perfil de fora pra dentro (*out-in*) da imagem, ou seja, a posição do primeiro pixel preto encontrado no rad que parte da borda da imagem em direção ao centro. Analogamente, em 5, define-se o perfil de dentro para fora (*in-out*), que se relaciona à posição do primeiro pixel preto encontrado no rad que parte do centro em direção à borda da imagem.

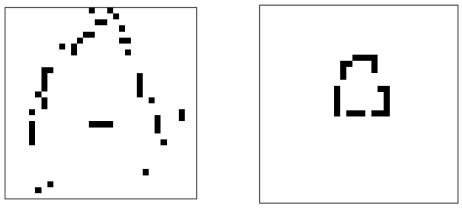
Um exemplo da aplicação do método utilizado pode ser visto na imagem 2, cujos exemplos foram retirados de [2].



(a) Imagem pré-processada (b) Histograma horizontal



(c) Histograma vertical (d) Histograma radial



(e) Perfil radial out-in (f) Perfil radial in-out

Fig. 2. Exemplo das características estruturais

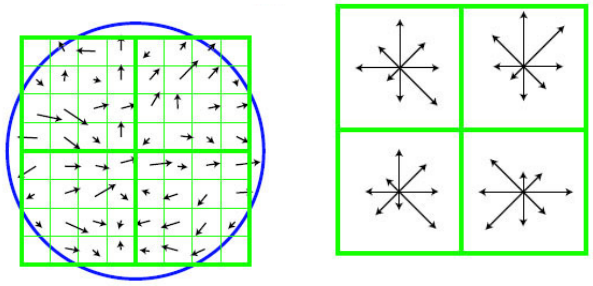
3.2. Histograma de gradientes orientados(HOG)

O HOG foi proposto primariamente com o intuito de resolver o problema de detecção de pedestres na rua em imagens estáticas, mas o método se mostrou muito útil nos mais diversos problemas da área de visão computacional. O seu funcionamento consiste em contar as ocorrências de uma determinada orientação do gradiente em certas porções da imagem, a que mais ocorrer naquele pedaço será considerada como o gradiente daquela partição da imagem. O conjunto de gradientes apresentados tem se mostrado uma boa forma de descrever o que ocorre na cena e tem apresentado ótimos resultados.

Em seu primeiro estágio, o algoritmo realiza uma normalização global da imagem, de forma a reduzir a influência de efeitos de iluminação na imagem, através de compressões gamma em cada canal de cor da imagem. Em seguida, calcula-se os gradientes de primeira ordem da imagem, que capturam contornos e algumas informações de textura. Daí parte-se para a divisão da imagem em células que combinarão os histogramas locais de uma dimensão de orientação de gradientes naquela região. Cada histograma de orientação divide a amplitude do ângulo do gradiente em um número predeterminado de bins. A magnitude dos gradientes dos pixels na célula são utilizados para a decisão da orientação do histograma. No próximo estágio, normaliza-se blocos selecionados das células obtidas até então de modo a aumentar a invariância de cada célula à iluminação às sombras e contraste. Os blocos de descritores então normalizados são denominados Histogramas de Gradientes Orientados, que são finalmente combinados em um vetor de características para posterior classificação. Mais detalhes sobre o Método de Histogramas de Gradientes Orientados podem ser vistos em [3].

Fig. 3. HoG aplicado a uma fotografia

A título de ilustração do método citado, na imagem 3, exemplifica-se a aplicação do HOG a uma fotografia. Em 3(a), a fotografia e sua conversão em histogramas de gradientes orientados em 3(b). Uma exemplificação da aplicação do método pode ser vista em 4, em que gradientes de diversas células da imagem são convertidas em blocos de histogramas de gradientes orientados.



(a) Gradientes das células da imagem (b) Blocos de histogramas de gradientes orientados

Fig. 4. Gradientes das células da imagem e sua conversão em histogramas de gradientes orientados

4. CLASSIFICADORES

4.1. Random Forest

Um dos classificadores utilizados nesse trabalho é o Random Forest. Esse método foi proposto por [4] e consistem em um conjunto de árvores de decisão construídas no momento de treinamento do método. Para construí-las são selecionados aleatoriamente alguns dos atributos contidos dentro do

vetor de características. Um vez feito isso, calcula-se a entropia apresentada por cada atributo e aquele que possuir a maior é escolhido para separar as classes naquela posição da árvore. A saída do classificador é dada pela classe que foi retornada como resposta pela maioria das árvores pertencentes à floresta. O método apresenta aprendizado não supervisionado, ou seja, dado às instancias de treinamento e as etiquetas que indicam a classe de cada uma, o algoritmo aprende a classifica-las sem que haja a intervenção de um usuário no processo. O método vem sendo largamente utilizado nas mais diversas áreas do aprendizado de máquina e vem apresentando ótimos resultados.

5. EXPERIMENTOS

Para verificar o desempenho do algoritmo, ele foi implementado e executado no MATLAB R2012b em um notebook Dell Inspiron 14z com processador Intel i5 1.70GHz, 6Gb de memória RAM e sistema operacional Windows 7 Home Premium x64.

Para o treinamento do algoritmo foi utilizado um conjunto balanceado que contém 7948 exemplos distribuídos entre as 36 classes para que o classificador aprenda a diferenciá-los de acordo com as características extraídas.

Foi utilizado um conjunto de validação que continha 2000 arquivos, que foi utilizado para melhorar o desempenho do algoritmo observando os resultados obtidos para ele e tomando ações que poderiam apresentar melhoras no resultado do algoritmo para esse conjunto.

Para testes foi utilizado um terceiro conjunto, que continha 2000 imagens. Vale ressaltar que não existe interseção entre os três conjuntos, logo nenhum elemento se encontra em mais que um conjunto.

Utilizou-se um conjunto de 500 árvores no classificador para determinar a classe das instâncias de entrada.

Existem duas instâncias de teste, uma que passa pelo pré-processamento e outra que não. Ambas divididas nos grupos descritos acima.

6. RESULTADOS

O algoritmo foi executado como foi dito na Seção 5, os resultados são apresentados a seguir na 1, sendo que o conteúdo das células representa a porcentagem média de acerto apresentada.

Table 1. Média de acerto(%)

	Sem pré-processamento	Com pré-processamento
HOG	63 03%	76 46%
Structural	54 34	75 25%
HOG & Structural	69 94%	80 25%

Apartir dos dados da tabela, nota-se que o Structural Characteristics apresentou um resultado médio ligeiramente pior quando comparado com o HOG, principalmente quando não se aplicou o pré-processamento, o que já era esperado dado que o Structural leva em consideração a distribuição e posição dos pixels fazendo com que ruídos atrapalhem considerávelmente a saída do extrator.

Ao analisar melhor os resultados, nota-se que o Structural Characteristics classificou razoavelmente a maioria dos caracteres, mas teve uma performance muito abaixo quando se olha os dígitos 0 e 6 e os caracteres Öë Ẅ. Já o HOG apesentou uma maior taxa de erro nos dígitos 0 e 8 e nos caracteres Öë Ẅ. O mal desempenhos de ambos no Ẅp̈rovavelmente se deve ao fato da má segmentação do caracter; já para o 0 e para o Ö, ambos confundiram esses dígitos entre sí devido a grande semelhança entre eles.

Quando os dois extratores são utilizados juntos, o resultado apresentado melhora um pouco, mas os caracteres 0, Öë Ẅc̈ontinuam apresentando uma acurácia de reconhecimento piores que as demais apresentando um melhora muito pequena quando comparada com a taxa de reconhecimento de cada extrator individualmente.

O melhor resultado obtido foi encontrado ao misturar os dois extratores junto com o pré-processamento, onde o reconhecimento de quase todos os caracteres apresentaram uma grande melhora.

7. CONCLUSÃO

Esse trabalho apresentou um sistema de reconhecimento de CAPTCHAS baseado no Random Forest. O sistema apresentou uma acurácia média de reconhecimento de dígitos relativamente baixa, 80 25% no melhor caso.

O algoritmo pode ser muito melhorado adicionando novas formas de extração de características ou testando o desempenho das já implementadas com outros classificadores como o SVM e o k-Means, por exemplo.

8. REFERENCES

[1] Jeff Yan and Ahmad Salah El Ahmad, "A low-cost attack on a microsoft captcha," in *Proceedings of the 15th ACM Conference on Computer and Communications Security*, New York, NY, USA, 2008, CCS '08, pp. 543–554, ACM.

[2] N. Fakotakis E. Kavallieratou, K. Sgarbas and G. Kokkinakis, "Handwritten word recognition based on structural characteristics and lexical support," in *Proceedings of the Seventh International Conference on Document Analysis and Recognition*. 2003, vol. 1, ICDAR 03.

[3] N. Dalal, "Histograms of oriented gradients for human detection," *Computer Vision and Pattern Recognition*, pp. 886–893, 2005.

[4] L. Breiman, "Random forest," *Machine Learning*, , no. 45, pp. 5–32, 2001.